DETEKSI SERANGAN *BLACK HOLE* TERHADAP PROTOKOL *ROUTING AD HOC ON-DEMAND DISTANCE VECTOR* MENGGUNAKAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Awit Priharsiwi
NIM: 145150201111067

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2021

PENGESAHAN

DETEKSI SERANGAN *BLACK HOLE* TERHADAP PROTOKOL *ROUTING AD HOC ON-DEMAND DISTANCE VECTOR* MENGGUNAKAN *SUPPORT VECTOR MACHINE*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

Awit Priharsiwi

NIM : 145150201111067

Skripsi ini telah diuji dan dinyatakan lulus pada

26 Juli 2021

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dany Primanita Kartikasari, S.T., M.Kom.	Fariz Andri Bakhtiar, S.T., M.Kom.
NIP. 197711162005012003	NIK. 201709840314 1001

Mengetahui
Ketua Jurusan Teknik Informatika

Achmad Basuki, S.T., M.MG., Ph.D.
NIP. 197411182003121002

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis dan diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No.20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 26 Juli 2021

Awit Priharsiwi
NIM / 145150201111067

KATA PENGANTAR

Puji syukur selalu penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat dan kasih saying-Nya, penulisan skripsi ini dapat penulis selesaikan. Sholawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW hingga di akhir masa.

Penulisan skripsi ini tidak dapat penulis selesaikan tepat waktu tanpa adanya dukungan morel dan materiel dari orang-orang terdekat penulis. Pada kata pengantar ini, penulis ingin menyampaikan rasa terima kasih kepada:

1. Allah SWT
2. Kedua orang tua yang secara ikhlas telah memberikan dukungan penuh baik dalam bentuk moril maupun materil.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M. T, Ph. D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Achamd Basuki, S. T, M. MG, Ph. D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Adhitya Bhwaiyuga, S. Kom, M. Sc selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Ibu Dany Primanita Kartikasari, S.T., M.Kom. dan Bapak Fariz Andri Bakhtiar, S.T., M.Kom. selaku dosen pembimbing skripsi penulis yang telah memberikan arahan, bimbingan serta dukungan hingga terselesaikannya penelitian ini.
7. Kedua adik saya, serta sahabat dan teman-teman yang tidak bisa saya sebutkan satu-persatu terimakasih atas *support*nya dalam penyelesaian penelitian ini.
8. Seluruh pihak yang telah memberikan dukungan dan bantuan selama penelitian ini berlangsung hingga dapat terselesaikan dengan baik.

Penulis menyadari bahwa dalam proses penulisan skripsi ini masih terdapat kesalahan dan kekurangan di beberapa bagian. Karena itu kritikan dan saran yang membangun sangat penulis harapkan demi perbaikan dan pengembangan skripsi ini di masa mendatang. Semoga skripsi ini bermanfaat bagi siapapun yang membaca dan menerapkannya dalam kehidupan sehari-hari.

Malang, 26 Juli 2021

Penulis

awit.priharsiwi@student.ub.ac.id

ABSTRAK

Awit Priharsiwi, Deteksi Serangan *Black Hole* Terhadap Protokol *Routing Ad Hoc On-Demand Distance Vector* Menggunakan *Support Vector Machine*

Pembimbing: Dany Primanita Kartikasari, S.T., M.Kom. dan Fariz Andri Bakhtiar, S.T., M.Kom.

Wireless Sensor Networks digunakan untuk mengumpulkan semua jenis data dan informasi dalam lingkungan yang kompleks. WSN menggunakan teknologi saluran komunikasi nirkabel untuk mengirimkan data, tanpa sarana perlindungan keamanan, data sangat rentan terhadap serangan internal dan eksternal. *Black hole* merupakan salah satu serangan yang terjadi pada WSN. Serangan *black hole* bekerja dengan menyatakan dirinya memiliki rute terpendek menuju *node* tujuan. Hal ini sangat membahayakan jika paket berisi informasi penting. Dari permasalahan tersebut, serangan *black hole* dapat terdeteksi dengan menggunakan metode *Support Vector Machine* pada protokol reaktif *Ad Hoc On-Demand Distance Vector.* SVM adalah metode yang ampuh untuk klasifikasi dan regresi. Tujuan penelitian ini untuk mengetahui akurasi SVM dalam mendeteksi serangan *black hole* dan berapa banyak data yang dibuang oleh *node* berbahaya ini. Pada hasil pengujian dapat dilihat ketika simulasi dengan kondisi normal, pada *packet loss* bernilai kecil (1), sedangkan simulasi terdapat *black hole* sebanyak (239) 100%. Simulasi dengan kondisi normal pada *packet delivery ratio* sebanyak 99.5816%, sedangkan simulasi terdapat *black hole* bernilai 0%. Simulasi dengan kondisi normal pada *average end-to-end delay* sebesar 202.429 ms, sedangkan simulasi terdapat *black hole* sekitar 172.524 ms hingga 214.979 ms. Hasil penelitian menunjukan bahwa algoritma SVM dalam pengklasifikasian serangan *black hole* dengan perhitungan menghasilkan hasil yang akurat.

Kata kunci: *Wireless Sensor Networks,* AODV*, Black hole*, *Support Vector Machine*.

ABSTRACT

Awit Priharsiwi, Detection of Black Hole Attacks on Ad Hoc On-Demand Distance Vector Routing Protocols Using Support Vector Machine

Supervisors: Dany Primanita Kartikasari, S.T., M.Kom. and Fariz Andri Bakhtiar, S.T., M.Kom.

Wireless Sensor Networks used to collect all of data and information in complex environments. WSN uses wireless communication channel technology to transmit data, without security protection, data is very vulnerable to internal and external attacks. Black hole is one of the attacks that occur on WSN. A black hole attack works by declaring itself to have the shortest route to the destination node. This is very dangerous if the package contains important information. From these problems, black hole attacks can be detected using the Support Vector Machine method on the Ad Hoc On-Demand Distance Vector reactive protocol. SVM is a powerful method for classification and regression. The purpose of this study is to determine the accuracy of SVM in detecting black hole attacks and how much data is discarded by these malicious nodes. In the test results, it can be seen when the simulation is under normal conditions, the packet loss is small (1), while in the simulation there are black holes (239) 100%. Simulation with normal conditions on the packet delivery ratio as much as 99,5816%, while the simulation has a black hole with a value of 0%. Simulations with normal conditions have an average end-to-end delay of 202,429 ms, while in the simulation there are black holes of about 172,524 ms to 214,979 ms. The results showed that the SVM algorithm in classifying black hole attacks with calculations produced accurate results.

Keywords: *Wireless Sensor Networks,* AODV*, Black hole, Support Vector Machine*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Wireless Sensor Network (WSN) alat utamanya adalah sensor. Penggunaan sensor dapat secara efektif dalam lingkungan eksternal dengan bantuan jaringan nirkabel untuk transmisi informasi untuk memenuhi kebutuhan pengguna. Perkembangan *Internet of things* tidak terlepas dari WSN. WSN dapat digunakan untuk mengumpulkan semua jenis data dan informasi dalam lingkungan yang kompleks, penerapan WSN mencakup semua aspek. Di bidang militer, dapat digunakan untuk mendeteksi penyebaran pasukan musuh dan situasi lainnya, dan memiliki fungsi mendeteksi polusi biologis dan kimia dan radiasi nuklir. Dalam hal pemantauan dan perlindungan lingkungan, dapat digunakan untuk pengumpulan data di lingkungan lapangan, pelacakan jejak hewan, menganalisis situasi pencemaran, dan memprediksi ledakan kebakaran hutan dan aliran puing-puing. Di bidang industri dan pertanian, pemantauan pertumbuhan tanaman dan produksi cerdas lainnya. Di bidang perawatan medis dan kesehatan, dapat mewujudkan fungsi memperoleh data fisiologis pasien, menganalisis kondisinya, dan menerima perawatan medis tepat waktu. Selain itu, WSN memainkan peran penting dalam bidang rumah pintar, transportasi pintar, kota pintar, perlindungan peninggalan budaya dan bisnis, dll (*Zhang Huanan, Xing Suping, Wang Jiannan, 2021*).

WSN menggunakan teknologi saluran komunikasi nirkabel untuk mengirimkan data, tetapi tanpa sarana perlindungan keamanan, data sangat rentan terhadap serangan internal dan eksternal (*Zhang Huanan, Xing Suping, Wang Jiannan, 2021*). Ada berbagai jenis serangan pada berbagai lapisan jaringan seperti *wormhole, sinkhole, selective forwarding, hello flood, acknowledgement flooding, false routing attacks,* serta *black hole* merupakan salah satu serangan yang terjadi pada WSN (*Gurjot Singh & Jagdeep Singh, 2013*).

Black hole selalu merespons setiap pesan RREQ dengan pesan RREP, bahkan ketika itu tidak memiliki rute aktual ke *node* tujuan. Ketika paket data mencapai titik *black hole*, paket lalu di *drop* alih-alih meneruskannya ke *hop* rute berikutnya. *Node* jahat yang menyiarkan pesan perutean dengan daya tinggi mampu menyesatkan sejumlah *node*. *Node-node* ini mencoba menggunakan *node* berbahaya sebagai *hop* berikutnya dalam rute mereka ke *sink node*. Tetapi *node* yang berada pada jarak yang jauh hanya akan mengirim pesan mereka dalam keadaan tidak sadar. Skenario yang sama, serangan *black hole* yang bertindak sebagai *node* berbahaya dapat meyakinkan semua *node* yang bertetangga, yang merupakan beberapa *hop* dari *sink node* bahwa mereka sebenarnya satu *hop* dari *node* tujuan. *Node-node* ini sebagai respons mencoba mengirim paket mereka langsung ke *node sink*. Akibatnya tidak ada paket yang dapat mencapai tujuan. (*Mehndi Samra & Naveen Kumar Gondhi, 2016*).

Protokol perutean dalam *Wireless Sensor Networks* dapat dikategorikan ke dalam tiga kategori, *yaitu table-driven/proaktif, on-demand/reaktif,* dan *hybrid*. Dengan menggunakan *routing protocol* reaktif (*on-demand)* seperti *Ad Hoc On-Demand Distance Vector (AODV)* yang memiliki kelebihan dalam menentukan jalur *routing* ketika diperlukan, meminimaliskan perutean yang tidak penting atau tidak dibutuhkan, sehingga node tidak perlu menyimpan satu atau lebih tabel yang berisi informasi *routing* membuat daya batrai hemat. *Routing protocol* reaktif merupakan protokol berbasis topologi dinamis, sehingga isi *routing table* mengikuti perubahan topologi dalam jaringan. Sedangkan *protocol routing* proaktif *(table driven)* tidak cocok untuk jaringan yang besar karena sering terjadinya perubahan *table* dalam jaringan yang mengakibatkan *node* tidak hemat *energy* dan *bandwith.* (*Sharma & Arunkumar, 2013*). Pada AODV terdapat dua proses utama yaitu proses *route discovery* dan proses *route maintenance*. Proses *route discovery* dilakukan ketika *node* tujuan akan mengirimkan data ke *node* lain dan tidak memiliki rute pada tabel *routing* dengan mengirim pesan *route request* (RREQ) kepada *node* tetangga dan akan mendapat balasan pesan *route replay* (RREP) ketika rute telah ditemukan. Sedangkan proses *route maintenance* dilakukan jika terjadi perubahan dan kerusakan rute dengan menggunakan pesan *route error* (RERR) (Dorri and Kamel, 2015)

Penelitian terdahulu dengan judul "*Attack Detection in Mobile Ad Hoc Networks Using SVM Algorithm*" menjelaskan saat *node* sumber menemukan jalur terpendek ke tujuan, kemudian mengirimkan paket eksperimental dan mengumpulkan data perilaku masing-masing *node* dalam transmisi paket ke tujuan. Dari data-data menunjukan perilaku *node* untuk membedakan *node* penyalahgunaan dan *node* aman. Hasil simulasi menunjukkan bahwa metode ini memiliki akurasi tinggi sekitar 95% (*Mohammad Zadeh M & Mirzaei Somarin A, 2017*). Penelitian lainnya dengan judul "*Preventing Black Hole Attack in Wireless Sensor Network Using HMM"* menjelaskan pendekatan pencegahan serangan *Black Hole* pada WSN menggunakan HMM. Pengukuran berdasarkan metrik seperti penundaan *end–to-end delay* dan *packets delivered ratio*. Algoritma berbasis HMM telah digunakan untuk memodelkan urutan keputusan jalur terpendek yang dipilih oleh *node* sumber untuk berkomunikasi dengan *node* tujuan. Sehingga memungkinkan pencegahan jalur dan *node* berbahaya, dan hasil eksperimen menunjukkan efisiensi terhadap pendekatan yang diusulkan untuk mencegah node berbahaya *(Hanane Kalkha, Hassan Satori, Khalid Satori, 2019*).

Berdasarkan latar belakang yang telah diuraikan sebelumnya, peneliti membuat rancangan lain untuk mendeteksi serangan *black hole* yaitu menggunakan metode *Support Vector Machine* pada protokol reaktif *Ad Hoc On-Demand Distance Vector (AODV)* untuk mengklasifikasikan paket yang tidak aman dengan mengetahui *node* mana saja yang terdapat *malicious* dan berapa banyak data yang dibuang oleh *node* berbahaya ini sehingga dapat terhindar dari serangan *black hole*. Diharapkan metode ini dapat menjadi menjadi pertimbangan dalam pencegahan serangan terutama *black hole*.

1.2 Identifikasi Masalah

Permasalahan yang timbul pada penelitian ini dapat dirumuskan menjadi:

1. Keamanan informasi pada *Wireless Sensor Networks* rentan terhadap serangan
2. Serangan *black hole* merupakan salah satu serangan yang terjadi pada WSN
3. Protokol perutean menggunakan *routing protocol* reaktif (*on-demand)* seperti *Ad Hoc On-Demand Distance Vector (AODV)*
4. Mendeteksi serangan *black hole* dengan menggunakan metode *Sensor Vector Machine* untuk mengklasifikasikan paket aman dan yang tidak aman

1.3 Rumusan Masalah

Adapun rumusan masalah yang akan dibahas pada penelitian ini meliputi:

1. Bagaimana dampak impelemntasi mekanisme untuk pendeteksian *black hole* terhadap kinerja protokol dari *packet loss, packet delivery ratio,* dan *average end-to-end delay* menggunakan AODV?
2. Bagaimana akurasi SVM dalam mendeteksi serangan *black hole* ?
3. Bagaimana pengaruh jumlah *node* terhadap akurasi SVM?

1.4 Tujuan

Mendapatkan algoritma yang tepat dan memiliki akurasi tinggi dalam mendeteksi serangan *black hole.*

1.5 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah membuktikan algoritma SVM dapat mendeteksi serangan *black hole* dengan akurat, sehingga dapat menjadi pertimbangan atau usulan untuk diterapkan di kehidupan nyata.

1.6 Batasan Masalah

1. Simulasi sistem menggunakan NS-2.35 untuk membangun skenario jaringan
2. *Node* yang disimulasikan yaitu 10,20,30,..100 *node*
3. Kecepatan pergerakan *node* 20,0-30,0 m/s
4. Pengujian dilakukan dengan parameter *average end to end delay, packet loss dan packet delivery ratio*

1.7 Sistematika Pembahasan

BAB I : **PENDAHULUAN**

Berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat peneltian, batasan masalah dan sistematika pembahasan.

BAB II : **LANDASAN KEPUSTAKAAN**

Berisi tentang kajian pustaka mengenai penelitian terdahulu yang terkait dengan topik penelitian. Pembahasaan dasar teori seperti, *Wireless Sensor Networks*, *routing protocol, Ad Hoc On-Demand Distance Vector, Support Vector Machine, Black Hole Attack,* parameter pengujian, dan *network simulator.*

BAB III : **METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang tahapan yang akan dilakukan yang terdiri dari studi literatur, analisis kebutuhan, perancangan, implementasi, pembahasan hasil pengujian, kesimpulan dan saran.

BAB IV : **HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil dari sisem berdasarkan penjelasan deskripsi umum sitem dan tujuan peneitian serta melakukan pembahasan berdasarkan perancangan yang dibuat.

BAB V : **PENUTUP**

Menjelaskan tentang kesimpulan dan saran. Kesimpulan dan jawaban dari rumusan masalah. Sedangkan saran untuk pengembangan lebih lanjut pada penelitian ini diharapkan akan menjadi lebih baik.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan membahas tentang beberapa teori yang terkait dengan penelitian.

2.1 Tinjauan Pustaka

Beberapa penelitian terdahulu yang dijadikan referensi oleh peneliti. Penelitian pertama yakni penelitian yang dilakukan oleh Mohammad Zadeh, M & Mirzaei Somarin, A pada tahun 2017 dengan mengatasi permasalahan serangan *black hole* pada protocol routing AODV di WSN yakni dengan metode untuk mempelajari mesin berdasarkan classifier SVM untuk mendeteksi prediksi *node* berbahaya menyerang WSN dan memperhitungkan karakteristik *node* akun dalam jaringan. Dengan mendeteksi *node* yang sehat dan destruktif, dapat memprediksi serangan di jalan. Karenanya, dengan prediksi serangan terhadap rute, memungkinkan ada rute yang aman dan rute yang tidak aman. Hasil simulasi menunjukkan bahwa metode yang diusulkan memiliki akurasi tinggi dalam mengklasifikasikan dan memprediksi *node* berbahaya dan berbahaya dalam jaringan. Ketepatan metode yang diusulkan adalah sekitar 95%, yang dapat dibandingkan dengan metode sebelumnya dalam memprediksi *node* berbahaya dan infiltrasi jaringan (Mohammad Zadeh M & Mirzaei Somarin A, 2017).

Penelitian kedua yang dijadikan referensi pendukung oleh peneliti yaitu penelitian yang dilakukan oleh Hanane Kalkha, Hassan Satori, Khalid Satori pada tahun 2019 dengan mengatasi permasalahan dengan menyajikan pendekatan pendekatan pencegahan serangan *Black Hole* pada WSN menggunakan HMM. Pengukuran berdasarkan metrik seperti penundaan *end–to-end delay* dan *packets delivered ratio*. Algoritma berbasis HMM telah digunakan untuk memodelkan urutan keputusan jalur terpendek yang dipilih oleh *node* sumber untuk berkomunikasi dengan *node* tujuan. Sehingga memungkinkan pencegahan jalur dan *node* berbahaya, dan hasil eksperimen menunjukkan efisiensi terhadap pendekatan yang diusulkan untuk mencegah node berbahaya (Hanane Kalkha, Hassan Satori, Khalid Satori, 2019).

Tabel 2.1 Tentang kajian pustaka yang memberikan informasi penelitian sebelumnya dan dasar teori yang berhubungan dengan topik penelitian.

Tabel 2.1 Kajian Pustaka

No	Nama Penulis, Tahun, dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	(Mohammad Zadeh, M., Mirzaei Somarin, A., 2017) *Attack Detection in*	Mendeteksi seragan *black hole* terhadap *routing*	Mendeteksi serangan pada MANET	Mendeteksi serangan pada WSN

	Mobile Ad Hoc Networks Using SVM Algorithm	*protocol reaktif* dengan Algoritma SVM		
2	(Hanane Kalkha, Hassan Satori, Khalid Satori, 2019) *Preventing Black Hole Attack in Wireless Sensor Network Using HMM*	Mendeteksi seragan *black hole* pada *routing protocol reaktif* pada WSN	Mencegah serangan menggunakan metode *HMM*	Mendeteksi serangan menggunakan metode *SVM Algorithm*

2.2 Dasar Teori

2.2.1 Wireless Sensor Networks

Wireless Sensor Networks (WSN) terdiri dari seperangkat perangkat sensor terbatas yang didistribusikan di ingkungan *indoor* atau *outdoor* tertentu. WSN bertujuan untuk mengumpulkan data lingkungan dan penempatan perangkat *node*. *Node* jaringan memiliki komunikasi aktual. Teknik pembentukan terpusat cocok untuk jaringan di mana kapasitas daya pemrosesan sebagian besar bergantung pada perangkat yang unik. Perangkat ini bertanggung jawab untuk pemrosesan, koordinasi, dan pengelolaan aktifitas informasi yang kemudian meneruskan data ke *sink node*.

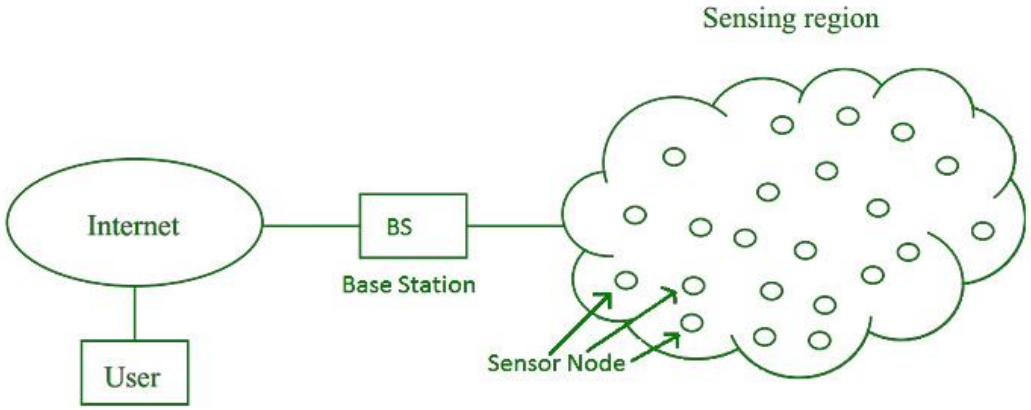
Dalam teknik formasi terdistribusi, informasi dikelola oleh setiap *node* dan keputusan diambil secara lokal dan terbatas pada lingkungannya (*single-hop neighbors*). Karakteristik utama dari jaringan terdistribusi meliputi:

1. Ada perangkat otonom
2. Setiap *node* membagikan informasi ke lingkungannya
3. Sangat cocok untuk aplikasi terdistribusi (sistem *multi-agen*, sistem yang diatur sendiri, dll)
4. Informasi diteruskan ke satu *node* tunggal
5. Perangkat interkoneksi (*router, bridge*, dll) tidak diperlukan
6. Fleksibilitasnya memungkinkan susah dalam penargetan lingkungan

Kompleksitas proses penerusan informasi membutuhkan algoritma yang kuat. Yang pertama harus memastikan pelaksanaan tugas-tugas tertentu dengan kinerja yang sebanding dengan solusi terpusat.

Protokol yang ditujukan untuk WSN terdistribusi harus dapat memberikan konsumsi energi yang efisien mempertimbangkan mobilitas *node*, kebisingan lingkungan, baterai terbatas, dan pesan-pesan yang lebih kecil di antara yang lain. Hal ini dapat terjadi karena teknik pengorganisasian dapat diklasifikasikan ke

dalam salah satu kelompok yang dibahas terpusat atau didistribusikan. (Miriam Carlos Mancilla, Ernesto López Mellado dan Mario Siller, 2016).



Gambar 2.1 *Wireless Sensor Network (WSN)*

Sumber: Anikakapoor (2021)

Gambar 2.1 di atas gambaran WSN dimana *node-node* berkomunikasi di antara mereka sendiri, terdapat *base station* guna untuk mengumpulkan data yang diminta atau diperlukan sehingga dapat di *forward* ke *user* melalui internet.

2.2.2 *Routing Protocol*

Routing protocol merupakan aspek penting sebuah jaringan dimana aspek tersebut sangatlah berpengaruh dalam kinerja sebuah jaringan. *Routing* merupakan suatu mekanisme penetuan jalur komunikasi dari *node* sumber ke *node* tujuan. *Protocol routing* mempunyai tugas untuk memberikan jalur terbaik menuju *node* tujuan dengan membentuk suatu tabel routing.

1. Routing Protokol Reaktif (*on demand)*

Dalam jenis *routing protocol* ini, rute dibuat hanya jika diperlukan. Dengan kata lain, ketika sebuah paket akan dikirim dari suatu sumber ke suatu tujuan, ia akan memanggil prosedur penemuan rute. Rute tetap berlaku sampai tujuan tercapai atau rute tidak lagi diperlukan. Contoh : DSR (*Dynamic Source Routing*) dan AODV (*Ad Hoc On-Demand Distance Vector*) . (Alslaim , et al., 2014)

2. Routing Protokol Proaktif (*table driven)*

Dalam tipe *routing protocol* ini, setiap *node* menyimpan satu atau lebih tabel yang berisi informasi routing ke setiap *node* lain dalam jaringan. Semua *node* terus memperbarui tabel routing mereka untuk mempertahankan tampilan terbaru dari jaringan. Contoh: DSDV (*Destination-Sequenced Distance-Vector*) dan WRP (*Wireless Routing Protocol*) (Alpasha, et al., 2014).

2.2.3 *Ad Hoc On-Demand Distance Vector* (AODV)

AODV (*Ad-hoc On-demand Distance Vector*) adalah protokol perutean reaktif yang sederhana, efisien, perutean *on-demand*. Algoritma ini dimotivasi oleh terbatasnya *bandwidth* yang digunakan untuk komunikasi nirkabel. Memperoleh

rute murni sesuai permintaan membuat AODV algoritma yang sangat berguna dan diinginkan. Setiap *node* seluler dalam jaringan bertindak sebagai router khusus dan rute diperoleh sesuai kebutuhan, sehingga membuat jaringan memulai sendiri.

1. *Control Packet*

Beberapa pesan yang digunakan dalam protokol routing AODV. Pesan-pesan ini digunakan untuk mengontrol proses penemuan rute dan pemeliharaan rute:

a. *Route Request Message (RREQ)*

Ketika *node* sumber ingin terhubung dengan *node* tujuan dan tidak memiliki entri rute ke *node* tujuan, paket control, bernama RREQ disiarkan oleh *node* sumber. RREQ berisi yang ditunjukkan pada Tabel 2.2.

Tabel 2.2 RREQ Format

Source Address
Request ID
Source Sequence No
Destination Address
Destination Sequence No
Hop Count

ID permintaan bertambah setiap kali *node* sumber mengirim RREQ baru. Pasangan (alamat sumber dan ID permintaan) mengidentifikasi RREQ secara unik. Saat RREQ bergerak dari satu *node* ke *node* lainnya, secara otomatis mengatur jalur mundur dari semua *node* ini kembali ke sumbernya. Setiap *node* yang menerima paket ini mencatat alamat *node* dari mana paket ini diterima. Proses ini disebut *Reverse Path Setup* (RPS).

b. *Route Request Message (RREQ)*

Jika sebuah *node* adalah tujuan atau memiliki rute yang valid ke tujuan, *unicasts* RREP kembali ke sumber. RREP memiliki format yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 RREP Format

Source Address
Destination Sequence
Destination Sequence No
Hop Count
Life Time

Saat *node* menerima pesan RREQ dari tetangga, ia mencatat alamat tetangga. Jadi, ketika *node* tujuan ditemukan, pesan RREP akan melakukan perjalanan di sepanjang jalur dan tidak ada lagi siaran yang diperlukan. Pesan RREP bergerak kembali ke sumber berdasarkan jalur sebaliknya yang dicatatnya. Saat RREP bergerak kembali ke sumber, setiap *node* di sepanjang jalur ini menetapkan *pointer* maju ke *node* dari tempat RREP menerima dan mencatat nomor urut tujuan terbaru ke tujuan permintaan. Proses ini disebut *Forward Path Setup* (FPS).

c. *Route Error Message (RERR)*
Semua *node* memantau lingkungan mereka sendiri. Ketika rute rusak atau tidak valid, RERR dihasilkan untuk memberi tahu *node* lain yang menggunakan rute ini, bahwa rute menjadi tidak valid. Pesan ini dibuat untuk menghindari pengiriman ulang oleh rute itu.

d. *HELLO Message*
Setiap *node* dapat mengetahui lingkungannya dengan menggunakan siaran lokal, yang disebut pesan HELLO. Tetangga *node* adalah semua *node* yang dapat langsung berkomunikasi dengan mereka. Pesan HELLO digunakan untuk memberi tahu para tetangga bahwa tautannya masih hidup

e. *Sequence Nmbers*
Nomor urut adalah fitur penting dari AODV untuk menentukan kesegaran informasi routing dan menjamin rute bebas *loop*. Nomor urut tujuan untuk setiap *node* tujuan disimpan dalam *table routing*, dan diperbarui ketika *node* menerima pesan dengan nomor urut yang lebih besar. Namun *node* meningkatkan nomor urut ketika:

- *Node* mengirim pesan RREQ, nomor urutnya bertambah.
- *Node* merespons pesan RREQ dengan mengirimkan RREP, nomor urut sendiri menjadi maksimum dari nomor urut saat ini dan nomor urut *node* dalam RREQ yang diterima.
- Sebuah *node* mengirim RERR untuk menunjukkan bahwa rute rusak.

Nomor urutan yang lebih tinggi adalah informasi yang lebih akurat, dan *node* mana pun yang mengirimkan nomor urutan tertinggi, informasinya dipertimbangkan dan rute dibuat di atas *node* oleh *node* lain.

2. *Route Discovery Process*

Ketika *node* sumber ingin mengirim paket data ke *node* tujuan, *node* sumber memeriksa dengan tabel peruteannya untuk menentukan apakah ada rute yang tersedia ke *node* tujuan. Jika demikian, node yang menggunakan rute ini untuk mengirim paket ke *node* tujuan. Dalam kasus di mana tidak ada rute ke *node* tujuan, proses penemuan rute dimulai dengan menyiarkan pesan RREQ. Setiap *node* memeriksa alamat sumber dan ID permintaan saat menerima pesan RREQ. Jika *node* telah menerima RREQ dengan alamat sumber dan ID permintaan yang sama, pesan RREQ yang baru akan dibuang. ID RREQ bertambah satu setiap kali

node sumber mengirim pesan RREQ. Permintaan Rute berisi nomor urut tujuan yang terakhir diketahui.

Jika *node* perantara memiliki entri rute untuk tujuan yang diinginkan dalam tabel peruteannya, ia membandingkan nomor urutan tujuan dalam tabel peruteannya dengan yang ada di pesan RREQ. Jika nomor urut tujuan dalam tabel peruteannya kurang dari yang ada di RREQ, itu menyiarkan ulang siaran RREQ ke tetangganya. Jika tidak, *unicasts* pesan balasan rute ke tetangganya dari mana ia menerima RREQ jika permintaan yang sama tidak diproses sebelumnya (ini diidentifikasi menggunakan ID permintaan dan alamat sumber). Berarti *node*nya adalah:

1. Siarkan RREQ dengan peningkatan *hop count* ke tetangganya (jika tidak ada entri rute untuk tujuan, atau ada satu tapi ini bukan rute yang terkini), atau
2. Kirim kembali RREP ke *node* sumber jika itu adalah *node* tujuan atau jika ia memiliki rute ke tujuan dengan nomor urut lebih besar atau sama dengan yang dari RREQ.

Pertukaran informasi rute akan diulang sampai RREQ mencapai *node* tujuan atau *node* menengah yang memiliki entri rute yang cukup baru untuk tujuan tersebut (Asma Ahmed, A. Hanan, Izzeldin Osman, 2015).

2.2.4 Support Vector Machine

SVM adalah metode yang ampuh untuk klasifikasi dan regresi. Operator ini mendukung jenis SVM C-SVC dan nu-SVC untuk tugas klasifikasi serta jenis SVM epsilon-SVR dan nu-SVR untuk tugas regresi. Selain itu, tipe *one-class* SVM didukung untuk estimasi distribusi. SVM mengambil sekumpulan data (input) dan memprediksi setiap masukan yang diberikan dari dua kelas menjadikan SVM pengklasifikasi linier biner non-probabilistik. Model SVM adalah representasi dari contoh-contoh sebagai titik-titik dalam ruang, yang dipetakan sedemikian rupa sehingga contoh-contoh dari kategori yang terpisah dibagi dengan celah yang jelas seluas mungkin. Contoh-contoh baru kemudian dipetakan ke dalam ruang yang sama dan diprediksi termasuk dalam kategori berdasarkan di sisi celah mana mereka berada.

Support Vector Machine membangun *hyperplane* atau kumpulan hyperplanes dalam ruang berdimensi tinggi atau tak terbatas, yang dapat digunakan untuk klasifikasi, regresi, atau tugas lainnya. Secara intuitif, pemisahan yang baik dicapai oleh hyperplane yang memiliki jarak terbesar ke titik data pelatihan terdekat dari semua kelas (disebut margin fungsional), karena secara umum semakin besar margin semakin rendah kesalahan generalisasi pengklasifikasi.

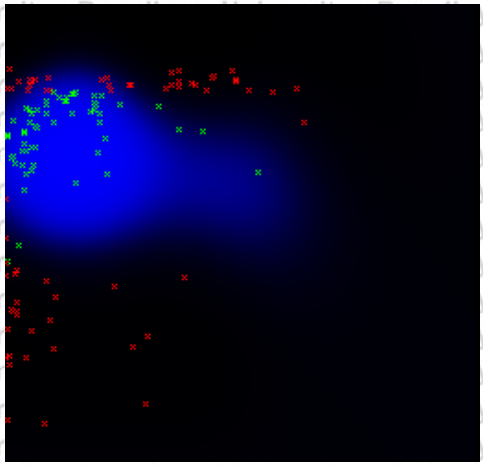
Berikut beberapa parameter yang digunakan untuk klasifikasi, regresi:

1. ***svm_type***

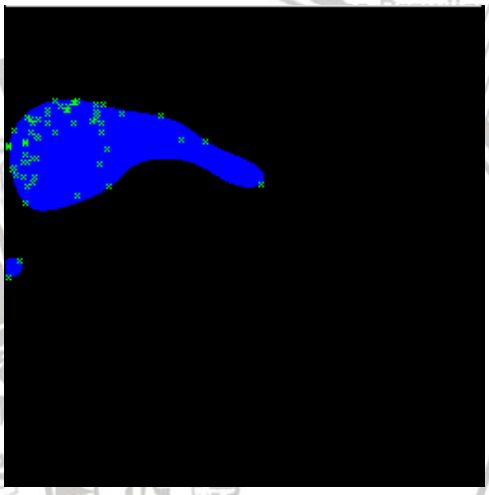
Jenis SVM dipilih melalui parameter ini. Operator ini mendukung tipe C-SVC dan nu-SVC SVM untuk tugas klasifikasi. Jenis SVM epsilon-SVR dan nu-SVR adalah untuk tugas regresi.
Type SVM: 0 = C-SVC, 1 = nu-SVC, 2 = *one-class SVM*

2. ***kernel_type***
Jenis fungsi kernel dipilih melalui parameter ini. Jenis kernel berikut ini didukung: linier, poli, rbf, sigmoid. Jenis kernel rbf adalah nilai default.
Type fungsi kernel:
Linier kernel
$$K(x,x_i) = x \cdot x^T$$
Polynominal kernel
$$K(x,x_i) = \left(1 + x \cdot x_i^T\right)^d$$
Radial Basis Function kernel
$$K(x,x_i) = e^{-\gamma \| x - x_i \|^2}$$
3. ***degree***
Parameter ini hanya tersedia jika parameter jenis kernel disetel ke 'poli'. Parameter ini digunakan untuk menentukan derajat fungsi kernel polinominal.
4. ***gamma***
Parameter ini hanya tersedia jika parameter jenis kernel disetel ke 'poly', 'rbf' atau 'sigmoid'. Parameter ini menetapkan gamma untuk fungsi kernel 'polinomial', 'rbf', dan 'sigmoid'. Nilai gamma mungkin memainkan peran penting dalam model SVM. Mengubah nilai gamma dapat mengubah akurasi model SVM yang dihasilkan. Jadi, merupakan praktik yang baik untuk menggunakan validasi silang untuk menemukan nilai gamma yang optimal.
5. **coef0**
Parameter ini hanya tersedia jika parameter jenis kernel disetel ke 'poly' atau 'precomputed'. Parameter ini menetapkan coef0 untuk fungsi kernel 'poly' dan 'precomputed'.
6. **C**
Parameter ini hanya tersedia jika parameter jenis svm disetel ke 'c-SVC', 'epsilon-SVR' atau 'nu-SVR'. Parameter ini menetapkan parameter biaya C untuk 'c-SVC', 'epsilon-SVR' dan 'nu-SVR'. C adalah parameter hukuman dari istilah kesalahan.
7. **nu**
Parameter ini hanya tersedia jika parameter jenis svm disetel ke 'nu-SVC', 'one-class' dan 'nu-SVR'. Parameter ini menetapkan parameter nu untuk 'nu-SVC', 'one-class' dan 'nu-SVR'. Nilainya harus antara 0,0 dan 0,5.

Berikut gambar menunjukkan model dalam ruang lingkungan yang dihasilkan dengan titik kehadiran yang sama tetapi dengan parameter yang berbeda. Karena SVM C-SVC membutuhkan titik absen, model pertama menyertakan satu set titik pseudo-absen yang dihasilkan secara acak di area (ruang lingkungan) yang jauh dari titik kehadiran:



Gambar 2.2 SVM C-SVC dengan parameter default.
(*Pseudo-absen* ditampilkan dengan warna merah)

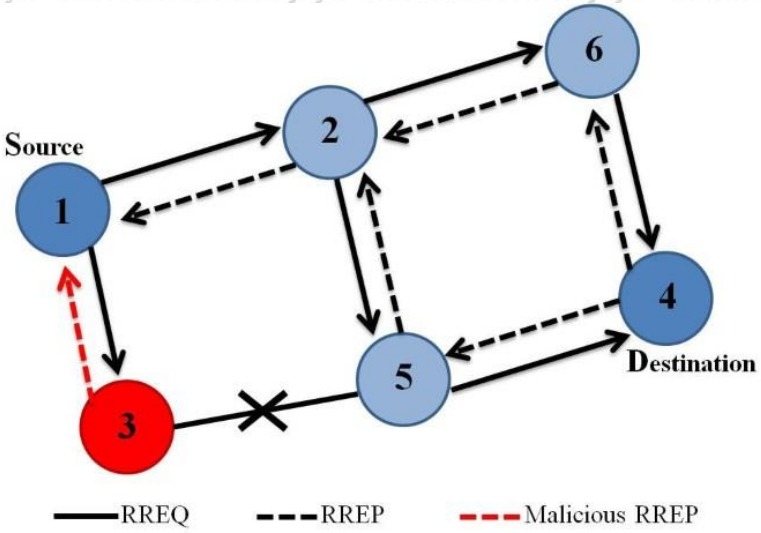


Gambar 2.3 SVM *one-class* dengan nu

2.2.5 Black Hole Attack

Dalam serangan *Black Hole*, *node* berbahaya mengiklankan dirinya sebagai jalur terpendek dan menarik semua lalu lintas data ke arahnya sendiri, lalu menyerap semua paket tanpa mengirimkannya ke tujuan. *Node* sumber memulai proses penemuan rute dengan menyiarkan RREQ ke tetangganya. Seluruh tetangga yang menerima RREQ meneruskannya ke arah tujuan dengan menambahkan alamat mereka. *Node* musuh mengirimkan RREP (dengan nomor urutan tertinggi dan jumlah *hop* minimum) sebagai respons terhadap *node* sumber sehingga berpura-pura sebagai *node* tujuan. Ketika *node* sumber menerima lebih dari satu respons, membandingkan nomor urut RREP yang diterima. Ia memilih jalur yang memiliki nomor urut terbesar. Jika keduanya RREP memiliki nomor urut yang sama, maka perhitungan *hop* minimum dipertimbangkan. Sebagai RREP dari *node* musuh memiliki *node* sumber urutan terbesar mengirim semua paket data

ke *node* itu. Oleh karena itu, *node* sumber dan *node* tujuan tidak dapat berkomunikasi satu sama lain. (Umashankar Ghugar & Jayaram Pradhan, 2017).



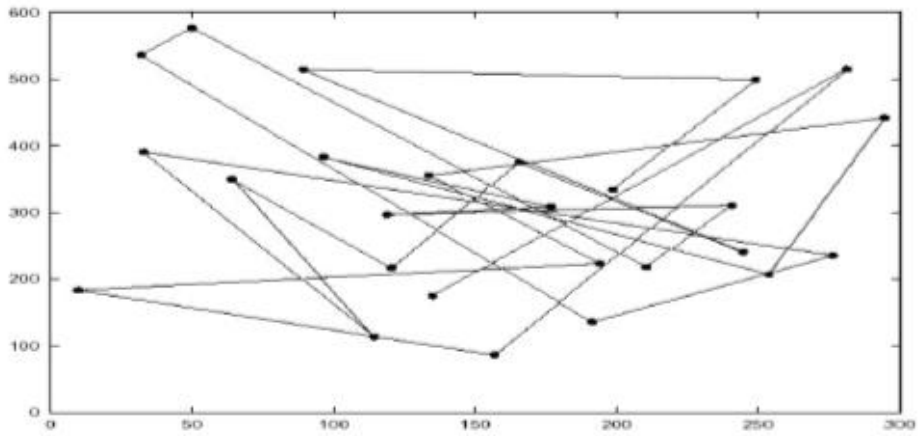
Gambar 2.4 Cara Kerja *Black Hole Attack*

Sumber: Fan-Hsun Tseng, dkk (2011)

Gambar 2.4 menampilakn *node*(1) adalah *source* dan *node*(4) adalah *destination.* Ketika *node*(1) mengirimkan paket data ke *node* (2) dan (3), proses penemuan rute dengan menyiarkan pesan RREQ ke *node-node* tetangga. Sehingga *node* lain pada gambar di atas menerima pesan ini. Kemudian terdapat *node*(3) berperan sebagai *node* jahat, Ini langsung mengirimkan pesan RREP palsu ke *node*(1) dengan nomor urut tertinggi serta *node*(2) juga mengirimkan pesan RREP yang sebenarnya ke *source node*(1) dengan urutan jumlah. Setelah paket diterima oleh *node*(3) atau *node* berbahaya paket kemudian dibuang yang seharusnya dikirimkan ke tujuan *node*(4).

2.2.6 Random Waypoint

Random Waypoint adalah model mobilitas *node* yang didistribusikan secara acak dalam jaringan. Prosedur yang ditempuh oleh RWP dalam memulai pergerakan adalah tiap *node* memilih tujuan secara acak (Mentari dkk.,2018). Setiap *node* akan menyebar secara bebas dan mempunyai kecepatan secara acak, sehingga tidak ada batasan yang dikenakan pada penyebarannya. Pada Gambar 2.3 digambarkan *Random Way Point*.



Gambar 2.5 Pergerakan *Random Waypoint*

Sumber: Rohankar, dkk (2012)

2.2.7 Parameter Pengujian

Berikut beberapa parameter yang akan digunakan untuk mengukur kinerja jaringan antara lain:

8. ***Packet Delivery Ratio*** merupakan perbandingan banyaknya jumah paket yang diterima oleh *node* penerima dangan total paket yang dikirimkan dalam suatu periode waktu tertentu (Husain, et al., 2010).

   Packet Delivery Ratio = $\frac{Jumlah\ paket\ yang\ diterima}{jumlah\ paket\ yang\ dikirim}$ X 100% (2.1)

9. ***Packet Loss*** adalah Jumlah total paket yang hilang, yang disebabkan *collision* dan *congestion* pada jaringan sehingga mengurangi efisiensi jaringan secara keseluruhan meski jumlah *bandwidth* cukup tersedia untuk aplikasi tersebut. (Pranata, 2016).

   Packet Loss = $\frac{paket\ data\ dikirim - paket\ data\ diterima}{paket\ data\ dikirim}$ X 100% (2.2)

10. ***Average End to End Delay*** merupakan rata-rata yang dibutuhkan oleh sebuah data untuk mencapai tujuan. Ini termasuk semua penundaan yang mungkin disebabkan oleh *buffering* selama *latency* penemuan *rute,* antrian pada antrian antarmuka, penundaan transmisi ulang pada MAC, dan *delay propagasi.* Metrik ini dihitung dengan mengurangi waktu di mana paket pertama ditransmisikan oleh sumber dari waktu di mana paket data pertama tiba ke tujuan (Husain, et al., 2010).

    Average End to End Delay = $\frac{Jumlahtotalwaktupengirimanpaket}{jumlahpaketyangterkirim}$ (2.3)

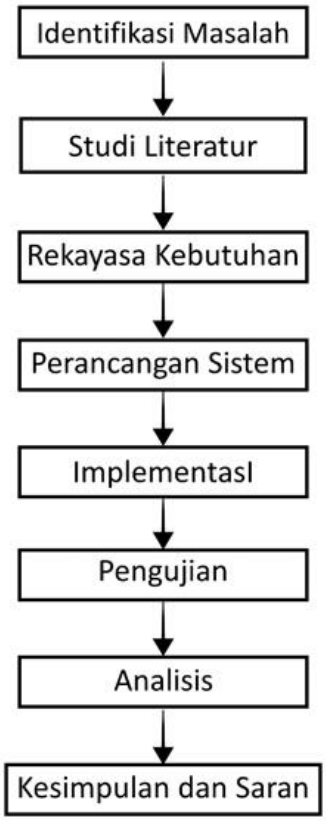
2.2.8 Network Simulator 2

Network simulator 2 mempunya *library* yang di dalamnya berisi *event scheduler, routing protocol*, dan *network component* yang disimulasikan oleh

pengguna. Terdapat 2 bahasa di dalam *network simulator* 2 yaitu C++ dan Bahasa Tcl yang digunakan pengguna untuk membuat *script*. Keunggulan dari Bahasa C++ yaitu mampu mendukung waktu simulasi yang cepat. Simulasi yang dimaksud simulasi dengan jumlah paket dan sumber data yang jumlahnya besar. Sebaliknya bahasa Tcl memberika respon yang lambat dari pada bahasa C++. Keunggulan dari bahasa Tcl yaitu jika terjadi kesalahan atau perubahan *script* respon yang diberikan bahasa Tcl dapat berlangsung dengan cepat dan interaktif (Issariyakul & Hossain, 2009).

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metodologi penelitian, adapun tahap-tahap penelitian dimulai dari mengidentifikasi studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian serta analisis, kesimpulan dan saran.



Gambar 3.1 Diagram Alur Penelitian

3.1 Perancangan Sistem

Pengumpulan sumber-sumber berupa teori yang berkaitan dengan *Routing protocol* AODV (*Ad Hoc On-demand Distance Vector*), *Support Vector Machine, Wireless Sensor Networks*, dan serangan *Black hole*. Pengumpulan sumber didapatkan dari berbagai buku, laporan penelitian, jurnal serta sumber lainnya yang dipelajari yaitu:

1. *Wireless Sensor Networks*

   Pada studi literatur, *Wireless Sensor Networks* mempelajari tentang bagaimana infrastruktur dalam jaringan

2. *Routing protocol* AODV (*Ad Hoc On-demand Distance Vector*)

   Pada studi literatur, AODV mempelajari tentang mekanisme kerja yang terdiri dari *route discovery* dan *route maintenance,* dan bagaimana penerapan protokolnya

3. Support Vector Machine

Pada studi literatur, *Support Vector Machine* merupakan metode dengan tujuan mendeteksi serangan berbahaya *black hole*

4. Serangan *Black Hole*

Pada studi literatur, serangan *Black Hole* merupakan serangan dengan tujuan pengambilan data antar *node* untuk selanjutnya akan di-*drop*

5. Network Simulator 2 (NS-2.35)

Pada studi literatur, NS-2.35 mempelajari bagaimana membuat simulasi *traffic* jaringan menggunakan *Network Simulator.* Komponen apa saja yang diperlukan untuk mendapatkan hasil simulasi yang baik. Pada *NS-2.35,* file tcl berfungsi sebagai file dijalan pada simulator dan hasil dari simulasi adalah file *tracefile*(.tr), dan *namfile*(.nam)

Perancangan sistem dan implementasi sistem menggunakan protokol AODV pada WSN. Pada penelitian ini terdiri dari kebutuhan fungsional dan kebutuhan non fungsional.

3.1.1 Kebutuhan Fungsional

Kebutuhan fungsioanl merupakan kebutuhan yang harus ada sehingga dapat berjalan guna memenuhi kebijakan sistem yang dibuat. Tabel 3.1 merupakan kebutuhan fungsional pada penelitian ini.

Tabel 3.1 Kebutuhan fungsional

No.	Kebutuhan Fungsional
1	Sistem dapat mengimplementasikan protokol *routing* AODV pada WSN
2	Sistem dapat mengimplementasikan serangan *black hole* terhadap AODV pada WSN.
3	Sistem dapat melakukan pendeteksian serangan *black hole* terhadap AODV pada WSN.

3.1.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional terbagi menjadi dua kebutuhan yaitu kebutuhan perangkat keras dan perangkat lunak. Proses implementasi pada sistem membutuhkan perangkat lunak. Kebutuhan perangkat lunak yang digunakan dalam implementasi pendeteksian serangan black hole terhadap protokol AODV pada WSN dapat dilihat pada Tabel 3.2

Tabel 3.2 Kebutuhan perangkat lunak

No	Perangkat	Keterangan
1	*Network Simulator 2 (NS2)*	Network Simulator berfungsi untuk membangun simulasi sistem dan menjalankan simulasi sistem.
2	*Ubuntu 14.04 LTS*	Ubuntu berfungsi sebagai lingkunan untuk menjalankan perangkat lunak network simulator dan network animator.

3	*NetAnim*	NetAnim berfungsi untuk menampilkan implementasi sistem secara visual dalam bentuk animasi.
4	*Sublime Text*	Sublime Text berfungsi untuk menulis code untuk implementasi sistem, serangan black hole, dan deteksi serangan black hole.

Kebutuhan perangkat keras yang digunakan dalam implementasi pendeteksian serangan black hole terhadap protokol AODV pada WSN dapat dilihat pada Tabel 3.3

Tabel 3.3 Kebutuhan perangkat keras

No	Perangkat	Keterangan
1	Laptop	Perangkat ini berfungsi untuk melakukan simulasi serangan dan pendeteksian *black hole* pada protokol AODV dan melakukan penyusunan dokumen penelitian. Perangkat memiliki spesifikasi, *processor* AMD A8, *Memory* 4 GB RAM, *Harddisk* 500 GB, VGA AMD R6,

3.2 Metode Evaluasi

Metode evaluasi dilakukan untuk memberikan gambaran implementasi protokol AODV pada WSN dan melihat kinerja dari protokol dalam kondisi normal. Berikut konfigurasi implementasi sistem menggunakan protokol AODV dapat dilihat pada Tabel 3.4 selama percobaan.

Tabel 3.4 Konfigurasi implementasi sistem

Parameter	Keterangan
Protokol *Routing*	AODV
Jumlah *Node*	10,20,30,40,50,60,70,80,90,100
Jumlah *Node Black Hole*	0 dan 2
Jenis Koneksi	TCP
Jenis Paket	CBR
Ukuran Paket	1024 *Bytes*
Ukuran *Rate* CBR	1000 Kb
Luas Area	1800 meter x 840 meter
Waktu Simulasi	40 detik
Tipe Mobilitas	*Random waypoint*
Kecepatan *Node*	20 sampai 30 m/s

3.2.1 Perancangan Serangan *Black Hole*

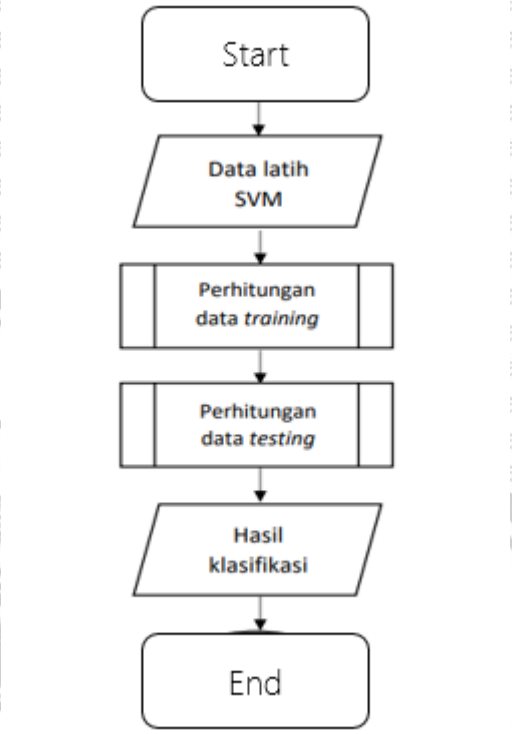
Berikut alur mekanisme *black hole* yang ditampilkan pada Gambar 3.1.

Gambar 3.2 Mekanisme *Black Hole*

Gambar 3.1 menjelaskan tentang rancangan serangan *black hole*. *Black hole* berperilaku seperti *node* biasa pada proses pembentukan *routing.* Setelah proses *routing* selesai dan jalur *routing* ke *node* tujuan terbentuk maka paket data dikirim. Paket data yang melalui *node black hole* di *drop* atau dibuang sehingga paket tidak sampai pada *node* tujuan.

3.2.2 Perancangan Deteksi Serangan *Black Hole*

Berikut alur mekanisme deteksi serangan *black hole black hole* yang ditampilkan pada Gambar 3.2.



Gambar 3.3 Mekanisme Deteksi Serangan *Black Hole*

Gambar 3.2 menjelaskan tentang rancangan deteksi serangan *black hole*. Setelah didapatkan data latih SVM lalu akan dilakukan perhitungan pada data latih, kemudian dilakukan lagi perhitungan pada data uji, sehingga menghasilkan kelas hasil klasifikasi.

3.2.3 Perancangan Parameter Pengujian

3.2.3.1 Packet Loss

parameter.awk

```
BEGIN {
sendLine = 0;
recvLine = 0;     }
{
# PDR
if (($1 == "s") &&  ( $7 == "cbr" ) && ( $4=="AGT" )) {
sendLine++; }
if (( $1 == "r") &&   ( $7 == "cbr" ) && ( $4=="AGT" ))    {
recvLine++; }
}
END {
printf "Packet Loss        \t= %d \n", (sendLine-recvLine); }
```

Pada script diatas merupakan rumus untuk menghitung nilai *packet loss*. Pada baris 1-3 inisialisasi dari nilai *sendLine* dan nilai *recvLine.* Baris 6-7 kondisi dimana

data dikirim apakah bernilai S dan memiliki tipe CBR dan memiliki tipe AGT yang kemudian disimpan pada variabel *sendLine*. Baris 8-9 kondisi dimana data dikirim apakah bernilai S dan memiliki tipe CBR dan memiliki tipe AGT yang kemudian disimpan pada variabel *recvLine*. Baris 12 menampilkan hasil dari perhitungan *packet loss*.

3.2.3.2 Packet Delivery Ratio

parameter.awk

```
BEGIN {
  sendLine = 0;
  recvLine = 0;      }
  {
# PDR
if ((($1 == "s") &&  ( $7 == "cbr" ) && ( $4=="AGT" )) {
sendLine++; }
if (( $1 == "r") &&  ( $7 == "cbr" ) && ( $4=="AGT" ))  {
recvLine++; }
  }
END {
 printf "Packet Delivery Ratio \t= %.4f \n", (recvLine/sendLine);
}
```

Pada script diatas merupakan rumus untuk menghitung nilai *packet delivery ratio*. Pada baris 1-3 inisialisasi dari nilai *sendLine* dan nilai *recvLine*. Baris 6-7 kondisi dimana data dikirim apakah bernilai S dan memiliki tipe CBR dan memiliki tipe AGT yang kemudian disimpan pada variabel *sendLine.* Baris 8-9 kondisi dimana data dikirim apakah bernilai S dan memiliki tipe CBR dan memiliki tipe AGT yang kemudian disimpan pada variabel *recvLine.* Baris 12 menampilkan hasil dari perhitungan *packet delivery ratio*.

3.2.3.3 Average End to End Delay

parameter.awk

```
BEGIN {
    seqno = -1;
    count = 0;
}
{
if($4 == "AGT" && $1 == "s" && seqno < $6) {
        seqno = $6;
#end-to-end delay
    if($4 == "AGT" && $1 == "s") {
  start_time[$6] = $2;
} else if( ($4 == "AGT") && ($7 == "cbr") && ($1 == "r")) {
        end_time[$6] = $2;
    } else if($1 == "D" && $7 == "cbr") {
        end_time[$6] = -1;
    }
}
END {
    for(i=0; i<=seqno; i++) {
        if(end_time[i] > 0) {
            delay[i] = end_time[i] - start_time[i];
            count++;
        }
        else
        {
            delay[i] = -1;
        }
    }
```

```
for(i=0; i<=seqno; i++) {
        if(delay[i] > 0) {
                n_to_n_delay = n_to_n_delay + delay[i];
        }
}
n_to_n_delay = n_to_n_delay/count;
print "Average End to End Delay     = " n_to_n_delay * 1000 "
ms";
}
```

Pada script diatas merupakan rumus untuk menghitung nilai *average end to end delay*. Baris 1-4 inisialisasi dari nilai *variable* seqno dan nilai count. Baris 6-7 berfungsi untuk mendapatkan variable seqno. Bertujuan untuk mendapatkan *sequence number* paket yang terakhir dikirim. Baris 9-10 untuk mengecek baris pada kolom *network trace file,* yaitu kolom ke 4 dan ke 1 apakah bernilai s, dan AGT. Baris 11-12 kondisi untuk mengecek baris pada kolom *network trace file,* yaitu kolom ke 4, 7, dan ke 1 apakah bernilai AGT, CBR, dan r. Baris 13-14 kondisi untuk mengecek baris pada kolom *network trace file,* yaitu kolom ke 1 dan ke 7 apakah bernilai D dan CBR. Baris 17-21 akhir dari perngumpulan data yang ditandai denga END. Proses perhitungan estimasi waktu paket yang terkirim dilakukan dengan mengurangi waktu *end_time* dengan *start_time*. Baris 23-25 untuk mengisi paket berstatus delay dengan nilai -1. Baris 28-31 menjumlahkan seluruh waktu paket terikirim yang disimpan pada variable *array delay*. Baris 34-36 menampilkan hasil rata-rata waktu *end-to-end delay* dengan membagi jumlah hasil total delay.

3.3 Implementasi

Implementasi terbagi menjadi beberapa tahapan. Pada langkah awal dilakukan pembuatan *node* berjumlah 10,20,30,..100 dengan 2 *node black hole*. Penempatan *node* dilakukan secara acak menggunakan pergerakan random waypoint. *Node* berada pada area 1800 meter x 840 meter. Pembuatan *node* dapat dilihat pada script dibawah.

```
1   set val(chan)      Channel/WirelessChannel     ;# Channel Type
2   set val(prop)      Propagation/TwoRayGround    ;# radio-propagation
3   model
4   set val(netif)     Phy/WirelessPhy             ;# network interface
5   type
6   set val(mac)       Mac/802_11                  ;# MAC type
7   set val(ifq)       Queue/DropTail/PriQueue     ;# interface queue type
8   set val(ll)        LL                          ;# link layer type
9   set val(ant)       Antenna/OmniAntenna         ;# antenna model
10  set val(ifqlen)    50                          ;# max packet in ifq
11  set val(nn)        20                          ;# number of mobilenode
12  set val(rp)        AODV                        ;# routing protocol
13  set val(x)         1800
14  set val(y)         840
15  set val(stop)      40.0
```

Penjelasan:

1. Baris 1-10 inisialisasi dari tipe *channel*, model propagasi yang digunakan, tipe jaringan *interface*, tipe *mac*, tipe *queue interface*, tipe *link layer*, model antenna yang digunakan, dan jumlah ukuran paket *node* yang digunakan.
2. Baris 11 inisialisasi dari jumlah mobile *node* yang akan digunakan yakni 7

3. Baris 12 inisialisasi dari protokol *routing* yang akan digunakan.
4. Baris 13-14 mendefinisikan luas area jaringan sumbu X dan sumbu Y.
5. Baris 15 mendefinisikan waktu simluasi akan berakhir

3.3.1 Implementasi Serangan *Black hole*

Implementasi serangan black *hole akan* dijalankan dengan menambahkan beberapa fungsi pada protokol AODV. Berikut merupakan beberapa fungsi yang ditambahkan.

```
1   ##aodv.h
2   bool malicious;
3   ##aodv.cc
4   malicious = false;
5   ##
6   if (strcasecmp(argv[1], "malicious") == 0) {
7
8           malicious = true;
9
10                  return TCL_OK;
11      }
12  ##
13  if (malicious == true ) {
14
15                  drop(p,  DROP_RTR_BLACK_HOLE); //BLACK_HOLE
16
17                  return;
18          }
19  ##cmu-trace.h
20  #define DROP_RTR_BLACK_HOLE         "HOLE"
21  ##Black Hole.tcl
22  $ns at 0.0 "[$n3 set ragent_] Black Hole"
23  $ns at 0.0 "[$n7 set ragent_] Black Hole"
```

Penjelasan:

1. Baris 2 mendefinisikan variable malicious.
2. Baris 4 inisialisasi ini diperlukan agar seluruh *node* dalam jaringan tidak memiliki *black hole* dalam inisialisasi awalnya.
3. Baris 6-10 proses string compare pada nilai argv[1] dengan char malicious, jika nilainya 0 maka variable malicious akan bernilai true dan tcl dapat di jalankan.
4. Baris 13-17 jika malicious bernilai true maka *drop* seluruh paket yang masuk ke dalam agen malicious dengan alasan sebagai *Black hole*.
5. Baris 19 mendefinisikan variable DROP_RTR_BLACK HOLE
6. Baris 21-22 mendefinisikan *node* 3 dan 7 sebagai agen dari *malicious*.

3.3.2 Implementasi Data Latih SVM

Implementasi algoritma SVM yang akan dijalankan fungsi sebagai berikut.

```
1   my $filename = 'hasil/Black Hole/hasil-bh.tr';
2   open(Trace,    $filename)    or    die    "Cannot    open    the
3   '$filename'\n";
4   while(<Trace>){
5       my @line = split;
6       if ( scalar(@line)>= 23){
7         my $layer = sprintf("%s", $line[3]);
```

```
        my $payload = sprintf("%s", $line[6]);
        my $sizePayload = sprintf("%s", $line[7]);
        my $RReq = sprintf("%s", $line[17]);
        my $RReqID = sprintf("%s",  $line[22]);
        my                                          @tmpData=
($layer,$payload,$sizePayload,$RReq,$RReqID,);
$filename = 'hasil/ori/hasil-ori.tr';
open(Trace,  $filename)  or  die  "Cannot  open  the
'$filename'\n";
while(<Trace>){
    my @line = split;
    if ( scalar(@line)>= 23){
        my $layer = sprintf("%s", $line[3]);
        my $payload = sprintf("%s", $line[6]);
        my $sizePayload = sprintf("%s", $line[7]);
        my $RReq = sprintf("%s", $line[17]);
        my $RReqID = sprintf("%s",  $line[22]);
        my                                          @tmpData=
($layer,$payload,$sizePayload,$RReq,$RReqID);
my $svm = new Algorithm::SVM(Type   => 'C-SVC',
                             Kernel => 'polynomial',
                             Gamma  => 64,
                             C      => 8);
my @allTrainning = ();
for(my $i = 0; $i <= $#dataLatih; $i++){
    my @layVal = 0;
    my @idx =0;
    my $value = $dataLatih[$i]->[0];
    if (grep( /^$value$/, @fLayer ))   {
    @layVal =   (1 + firstidx { $_ eq $value } @fLayer) /
(scalar(@fLayer)+1);
    }else{
    }
    my @payVal = 0;
    @idx =0;
    $value = $dataLatih[$i ]->[1];
    if (grep( /^$value$/, @fpayload ))   {
    @payVal =  (1 + firstidx { $_ eq $value } @fpayload)  /
(scalar(@fpayload)+1);
    }else{
    }
    my @pkSize =  ($dataLatih[$i ]->[2] + 0) / 100;
    @idx =0;
    my @rrVal = 0;
    $value =   $dataLatih[$i]->[3];
    @rrVal =   (1 +   firstidx { $_ eq $value } @fRReq) /
(scalar(@fRReq)+1);
    @idx =0;
    my @rrIDVal = 0;
    $value = $dataLatih[$i ]->[4];
my $jmlError=0;
my $jmlBenar=0;
for(my $i = 0; $i <= $#allTrainning; $i++){
    my @values = $allTrainning[$i] ;
    my $dTrain = new Algorithm::SVM::DataSet(Label => 1,
                                             Data  =>  @values);
    my $resTrain = 0+$svm->predict($dTrain);
    if ($resTrain eq $labelLatih[$i]) {
```

```
66          $jmlBenar+=1;
67      }else {
68          $jmlError+=1;
69      }
70  }
71  $svm->load('mysvm.model');
72  my $akurasi = ($jmlBenar / ($jmlBenar+$jmlError)) *100;
73  print "\n\nAKURASI : $akurasi%\n";
74  print "DONE\n";
```

Penjelasan:

1. Baris 1 inisialisasi filename untuk folder hasil *black hole*
2. Baris 2 membuka file *tracefile* jika gagal maka ada pesan lain mucul
3. Baris 6-13 mengambil data dari file *trace* untuk dijadikan data latih seperti pada bagian *layer, payload, sizePayload, RReq, RReqID*.
4. Baris 14 inisialisasi filename untuk folder hasil normal
5. Baris 15 membuka file *tracefile* jika gagal maka ada pesan lain mucul
6. Baris 19-26 mengambil data dari file *trace* untuk dijadikan data latih seperti pada bagian *layer, payload, sizePayload, RReq, RReqID*.
7. Baris 27-30 inisialisasi algoritma svm (type, kernel, gamma, C)
8. Baris 31-57 inisialisasi data latih svm
9. Baris 58 JmlError bernilai 0
10. Baris 59 JmlBenar bernilai 0
11. Baris 60 melakukan perulangan dari 0 dengan pertambahan 1 setiap perulangan
12. Baris 61-70 melakukan data latih
13. Baris 71 memuat data dari mysvm.model
14. Baris 72 perhitungan akurasi
15. Baris 73-74 menampilkan hasil % akurasi

3.3.3 Implementasi Deteksi Serangan *Black hole*

Implementasi Deteksi serangan *black hole* dengan Algoritma SVM. Berikut beberapa fungsi.

```
1   my $jml Black Hole=0;
2   my $jmlTrue=0;
3   for(my $i = 0; $i <= $#allUjiData; $i++){
4       my @values = $allUjiData[$i] ;
5       my $dTrain = new Algorithm::SVM::DataSet(Label => 1,
6                                                Data  => @values);
7       my $resTrain = 0+$svm->predict($dTrain);
8       if ($resTrain eq 1 ) {
9           $jmlTrue+=1;
10      }else {
11          $jml Black Hole+=1;
12      }
13  }
14  my $jmlPaket = $jmlTrue+$jml Black Hole;
15  my $prosen = 100 - (($jmlTrue / ($jmlTrue+$jml Black Hole))
16  *100);
17  if ($jml Black Hole > 0) {
18  y $jmlPaket = $jmlTrue+$jml Black Hole;
19
```

```
20  my $prosen = 100 - (($jmlTrue / ($jmlTrue+$jmlBlack Hole))
    *100);
21  print "JML PAKET : $jmlPaket\n";
22  print "PAKET BLACK HOLE : $jmlBlack Hole\n";
23  print "PAKET NORMAL : $jmlTrue\n";
24  print "\nProsentasi Paket Black Hole : $prosen%\n";
25  print "DONE\n";
```

Penjelasan:

1. Baris 1 jml*Black Hole* bernilai 0
2. Baris 2 jml*True* bernilai 0
3. Baris 3 melakukan perulangan dari 0 dengan pertambahan 1 setiap perulangan
4. Baris 4-13 melakukan data uji
5. Baris 14-20 perhitungan data dari jumlah paket yang tidak terdapat dan paket yang terdapat *black hole*
6. Baris 21 menampilkan Jumlah keselurah paket
7. Baris 22 menampilakn jumlah paket yang telah di*drop* oleh *black hole*
8. Baris 23 menampilkan jumlah paket normal atau sisa dari paket sebelumnya
9. Baris 24 menampilkan persentase paket yang di*drop black hole*

BAB 4 HASIL DAN PEMBAHASAN

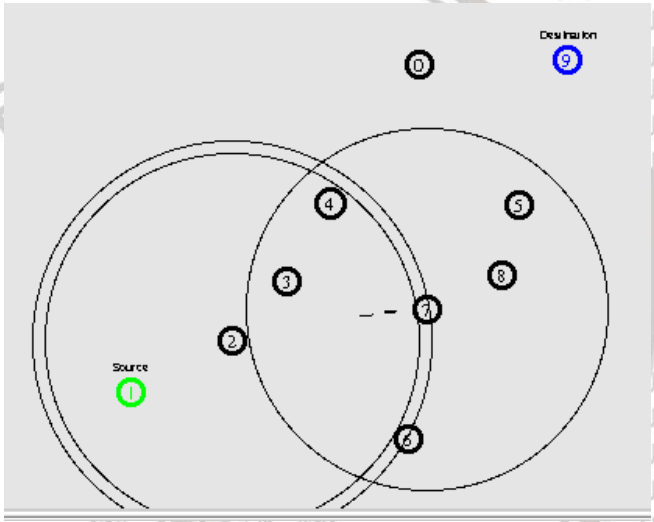
Pada bab ini mengulas tentang pengujian yang telah dijalankan serta melakukan evaluasi pada hasil pengujian.

4.1 Hasil Pengujian

Hasil pengujian ini diperoleh dari skenario yang telah dirancang di bab sebelumnya lalu diimpelemntasikan ke dalam system guna mengetahui apakah sudah berjlan sesuai yang diharapkan. Hasil pengujian dari serangan *black hole* terhadap AODV akan dibahas untuk mendapatkan kesimpulan. Berikut ini merupakan hasil pengujian yang telah dilakukan.

4.1.1 Hasil Pengujian *Node* Normal

Berikut adalah kegiatan *node* pada kondisi normal.



Gambar 4.1 Tampilan 10 *node* kondisi normal

Pada gambar 4.1 merupakan tampilan dari simulasi jaringan WSN dalam kondisi normal pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 10. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.1 ada 3 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data.

4.1.2 Hasil Pengujian Parameter *Node* Normal

Berikut adalah hasil parameter pengujian pada kondisi normal.

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/ori/h
asil-ori.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 238
Packet Loss =  1
Packet Delivery Ratio = 99.5816%
Average End-to-End Delay = 202.429 ms
Total Dropped Packets = 0
```

Gambar 4.2 Hasil parameter pengujian kondisi normal 10 *node*

Gambar 4.2 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* sebesar 99.5816%, *packet loss 1* dan *average end to end delay* 202.429 ms.

4.1.3 Hasil Pengujian Deteksi *Node* Normal

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole*.

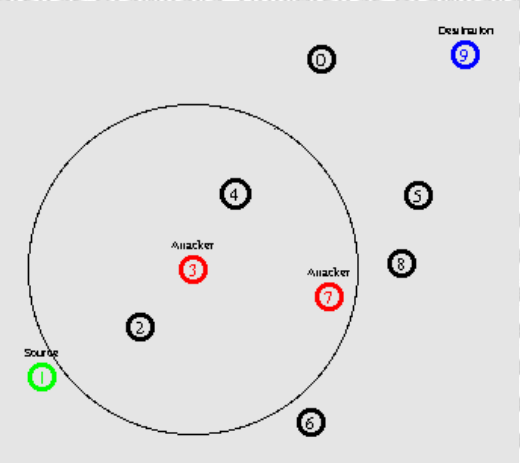
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/ori/hasil-o
ri.tr
LOAD 'hasil/ori/hasil-ori.tr'
JML PAKET : 228
PAKET BLACKHOLE : 0
PAKET NORMAL : 228

Prosentasi Paket Blackhole : 0%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.3 Hasil pengujian SVM 10 *node* normal

Pada Gambar 4.3 di atas menjelaskan jumlah paket normal sebanyak 228, dan tidak terdapat paket *black* hole, persentasi paket *black hole* sebanyak 0%.

4.1.4 Hasil Pengujian 10 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*



Gambar 4.4 Tampilan 10 *node* kondisi terdapat *black hole*

Pada gambar 4.4 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 10. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.4 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.5 Hasil Pengujian Parameter 10 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
=========================
   Parameter Pengujian
=========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 172.524 ms
Total Dropped Packets = 238
```

Gambar 4.5 Hasil parameter pengujian terdapat *black hole* 10 *node*

Gambar 4.5 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi terdapat *black hole* dengan menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* sebesar 172.524 *ms.*

4.1.6 Hasil Pengujian Deteksi Serangan *Black Hole* 10 *Node*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

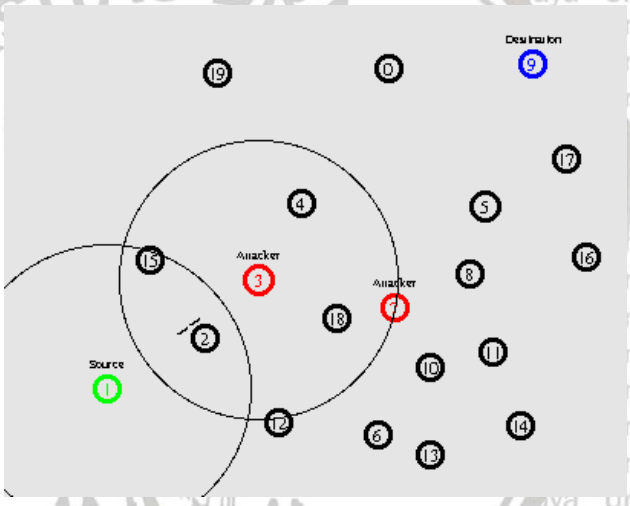
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/H
asil-bh.tr
JML PAKET : 3009
PAKET BLACKHOLE : 2856
PAKET NORMAL : 153

Prosentasi Paket Blackhole : 94.9152542372881%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.6 Hasil pengujian SVM *black hole* 10 *node*

Pada Gambar 4.6 di atas menjelaskan jumlah paket sebanyak 3009, yang dibuang oleh *black hole* sebanyak 2856, sisa paket normal sebanyak 153, persentasi paket *black hole* sebanyak 94.9152%.

4.1.7 Hasil Pengujian 20 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*



Gambar 4.7 Tampilan 20 *node* kondisi terdapat *black hole*

Pada gambar 4.7 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 20. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.7 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.8 Hasil Pengujian Parameter 20 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 172.817 ms
Total Dropped Packets = 238
```

Gambar 4.8 Hasil parameter pengujian terdapat *black hole* 20 *node*

Gambar 4.8 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi terdapat *black hole* dengan menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* sebesar 172.817 *ms.*

4.1.9 Hasil Pengujian Deteksi Serangan *Black Hole* 20 *Node*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

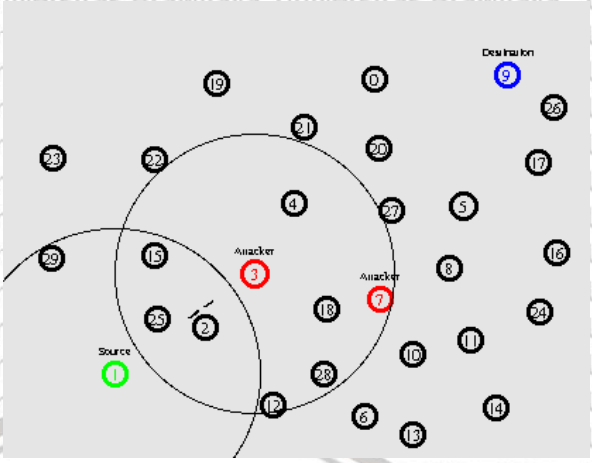
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 5211
PAKET BLACKHOLE : 4751
PAKET NORMAL : 460

Prosentasi Paket Blackhole : 91.172519669929%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.9 Hasil pengujian SVM *black hole* 20 *node*

Pada Gambar 4.9 di atas menjelaskan jumlah paket sebanyak 5211, yang dibuang oleh *black hole* sebanyak 4751, sisa paket normal sebanyak 460, persentasi paket *black hole* sebanyak 91.1725%.

4.1.10 Hasil Pengujian 30 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*



Gambar 4.10 Tampilan 30 *node* kondisi terdapat *black hole*

Pada gambar 4.10 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 30. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.10 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.11 Hasil Pengujian Parameter 30 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 172.526 ms
Total Dropped Packets = 238
```

Gambar 4.11 Hasil parameter pengujian terdapat *black hole* 30 *node*

Gambar 4.11 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* sebesar 172.526 *ms.*

4.1.12 Hasil Pengujian Deteksi Serangan *Black Hole* 30 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.

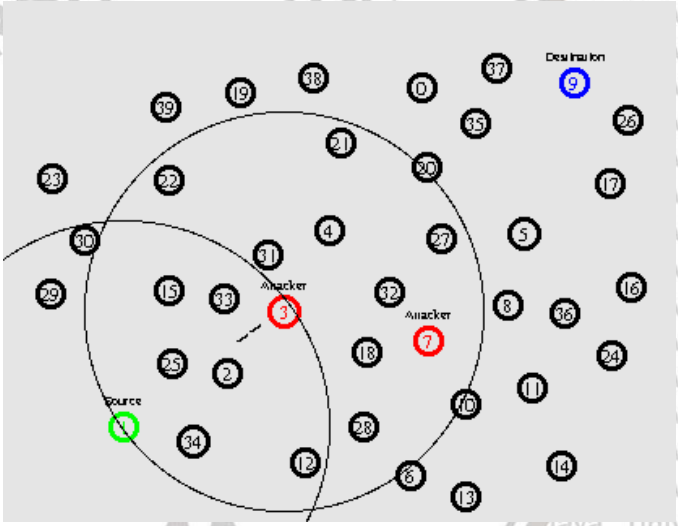
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 8493
PAKET BLACKHOLE : 7598
PAKET NORMAL : 895

Prosentasi Paket Blackhole : 89.4619098080772%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.12 Hasil pengujian SVM *black hole* 30 *node*

Pada Gambar 4.12 di atas menjelaskan jumlah paket sebanyak 8493, yang dibuang oleh *black hole* sebanyak 7598, sisa paket normal sebanyak 895, persentasi paket *black hole* sebanyak 89.4619%.

4.1.13 Hasil Pengujian 40 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.



Gambar 4.13 Tampilan 40 *node* kondisi terdapat *black hole*

Pada gambar 4.13 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 40. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.13 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.14 Hasil Pengujian Parameter 40 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 172.53 ms
Total Dropped Packets = 238
```

Gambar 4.14 Hasil parameter pengujian terdapat *black hole* 40 *node*

Gambar 4.14 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* sebanyak 172.53 *ms.*

4.1.15 Hasil Pengujian Deteksi Serangan *Black Hole* 40 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*

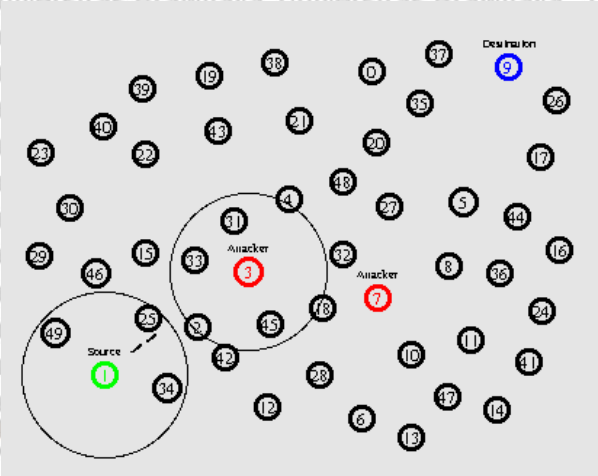
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 11489
PAKET BLACKHOLE : 9977
PAKET NORMAL : 1512

Prosentasi Paket Blackhole : 86.8395856906606%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.15 Hasil pengujian SVM *black hole* 40 *node*

Pada Gambar 4.15 di atas menjelaskan jumlah paket sebanyak 11589, yang dibuang oleh *black hole* sebanyak 9977, sisa paket normal sebanyak 1512, persentasi paket *black hole* sebanyak 86.8395%.

4.1.16 Hasil Pengujian 50 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.



Gambar 4.16 Tampilan 50 *node* kondisi terdapat *black hole*

Pada gambar 4.16 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 50. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.16 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.17 Hasil Pengujian Parameter 50 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole*.

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 214.691 ms
Total Dropped Packets = 239
```

Gambar 4.17 Hasil parameter pengujian terdapat *black hole* 50 *node*

Gambar 4.17 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* sebesar 214.691 *ms*.

4.1.18 Hasil Pengujian Deteksi Serangan *Black Hole* 50 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.

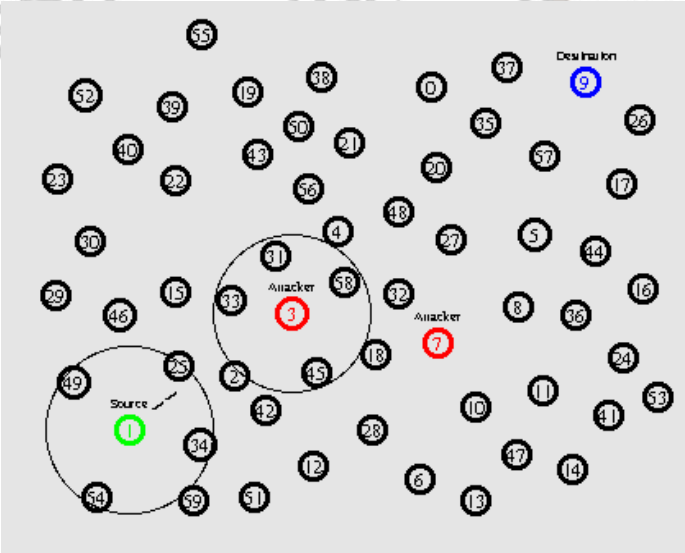
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 14741
PAKET BLACKHOLE : 12428
PAKET NORMAL : 2313

Prosentasi Paket Blackhole : 84.3090699409809%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.18 Hasil pengujian SVM *black hole* 50 *node*

Pada Gambar 4.18 di atas menjelaskan jumlah paket sebanyak 14741, yang dibuang oleh *black hole* sebanyak 12428, sisa paket normal sebanyak 2313, persentasi paket *black hole* sebanyak 84.3090%.

4.1.19 Hasil Pengujian 60 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.



Gambar 4.19 Tampilan 60 *node* kondisi terdapat *black hole*

Pada gambar 4.19 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 60. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.19 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.20 Hasil Pengujian Parameter 60 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 172.979 ms
Total Dropped Packets = 239
```

Gambar 4.20 Hasil parameter pengujian terdapat *black hole* 60 *node*

Gambar 4.20 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* 172.979 *ms.*

4.1.21 Hasil Pengujian Deteksi Serangan *Black Hole* 60 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*

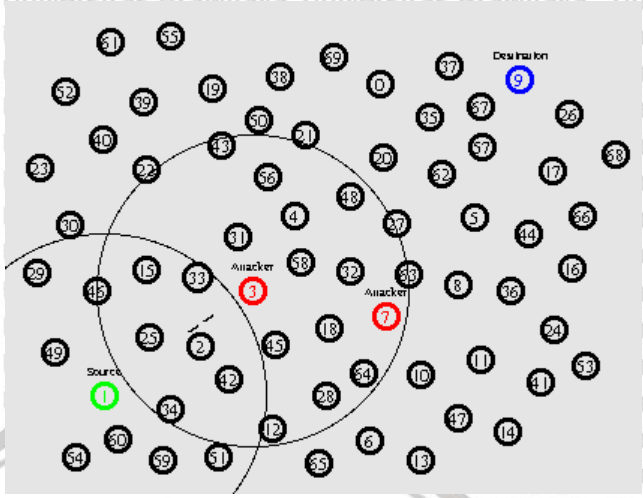
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 17839
PAKET BLACKHOLE : 14788
PAKET NORMAL : 3051

Prosentasi Paket Blackhole : 82.8970233757498%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.21 Hasil pengujian SVM *black hole* 60 *node*

Pada Gambar 4.21 di atas menjelaskan jumlah paket sebanyak 17839, yang dibuang oleh *black hole* sebanyak 14788, sisa paket normal sebanyak 3051, persentasi paket *black hole* sebanyak 82.8970%.

4.1.22 Hasil Pengujian 70 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*



Gambar 4.22 Tampilan 70 *node* kondisi terdapat *black hole*

Pada gambar 4.22 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 70. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.22 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.23 Hasil Pengujian Parameter 70 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 214.729 ms
Total Dropped Packets = 239
```

Gambar 4.23 Hasil parameter pengujian terdapat *black hole* 70 *node*

Gambar 4.23 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* 214.729 *ms.*

4.1.24 Hasil Pengujian Deteksi Serangan *Black Hole* 70 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.

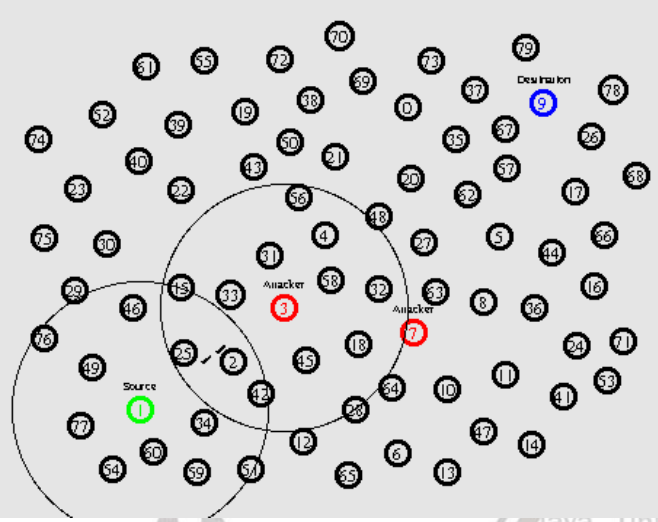
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 21113
PAKET BLACKHOLE : 16732
PAKET NORMAL : 4381

Prosentasi Paket Blackhole : 79.2497513380382%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.24 Hasil pengujian SVM *black hole* 70 *node*

Pada Gambar 4.23 di atas menjelaskan jumlah paket sebanyak 21113, yang dibuang oleh *black hole* sebanyak 16732, sisa paket normal sebanyak 4381, persentasi paket *black hole* sebanyak 79.2497%.

4.1.25 Hasil Pengujian 80 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole*.



Gambar 4.25 Tampilan 80 *node* kondisi terdapat *black hole*

Pada gambar 4.25 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 80. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.25 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.26 Hasil Pengujian Parameter 80 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 214.734 ms
Total Dropped Packets = 239
```

Gambar 4.26 Hasil parameter pengujian terdapat *black hole* 80 *node*

Gambar 4.26 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* 214.734 *ms.*

4.1.27 Hasil Pengujian Deteksi Serangan *Black Hole* 80 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*

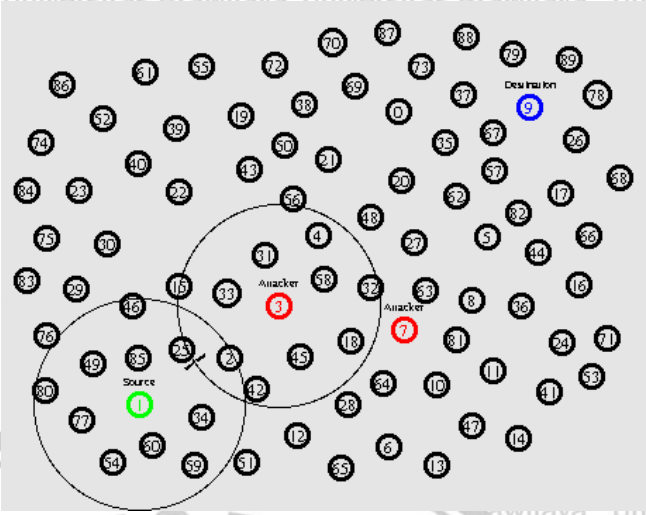
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 21418
PAKET BLACKHOLE : 16730
PAKET NORMAL : 4688

Prosentasi Paket Blackhole : 78.1118685218041%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.27 Hasil pengujian SVM *black hole* 80 *node*

Pada Gambar 4.27 di atas menjelaskan jumlah paket sebanyak 21418, yang dibuang oleh *black hole* sebanyak 16730, sisa paket normal sebanyak 4688, persentasi paket *black hole* sebanyak 78.1118%.

4.1.28 Hasil Pengujian 90 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*



Gambar 4.28 Tampilan 90 *node* kondisi terdapat *black hole*

Pada gambar 4.28 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 90. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.28 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.29 Hasil Pengujian Parameter 90 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 214.783 ms
Total Dropped Packets = 239
```

Gambar 4.29 Hasil parameter pengujian terdapat *black hole* 90 *node*

Gambar 4.29 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* 214.783 *ms.*

4.1.30 Hasil Pengujian Deteksi Serangan *Black Hole* 90 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*

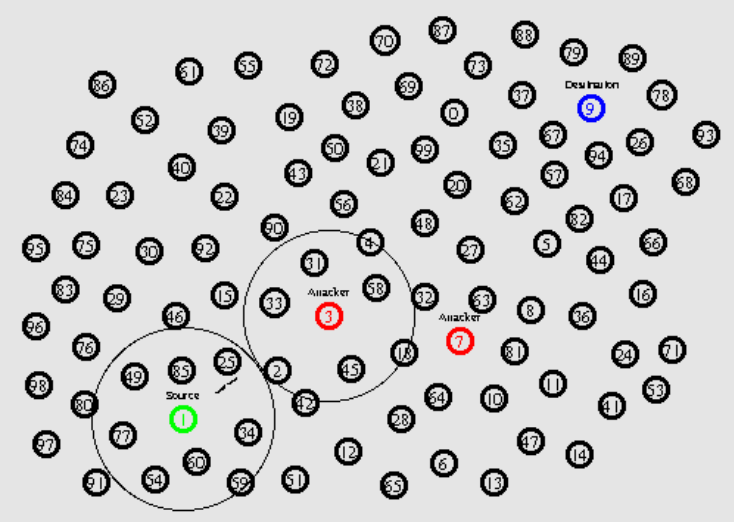
```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 24045
PAKET BLACKHOLE : 17686
PAKET NORMAL : 6359

Prosentasi Paket Blackhole : 73.5537533790809%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.30 Hasil pengujian SVM *black hole* 90 *node*

Pada Gambar 4.30 di atas menjelaskan jumlah paket sebanyak 24045, yang dibuang oleh *black hole* sebanyak 17686, sisa paket normal sebanyak 6359, persentasi paket *black hole* sebanyak 73.5537%.

4.1.31 Hasil Pengujian 100 *Node* Terdapat *Black Hole*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*



Gambar 4.31 Tampilan 100 *node* kondisi terdapat *black hole*

Pada gambar 4.31 merupakan tampilan dari simulasi jaringan WSN dalam kondisi terdapat *black hole* pada aplikasi NetAnim. Simulasi menggunakan protokol AODV dengan *node* yang berjumlah 100. Posisi setiap *node* telah ditentukan pada area seluas 1800 meter x 840 meter. Pada Gambar 4.31 ada 4 macam warna *node* dan memiliki peran yang berbeda. *Node* dengan warna hitam sebagai *node* tetangga yang digunakan sebagai perantara pengiriman data. *Node* berwarna hijau (1) sebagai *node* sumber yang berperan untuk melakukan pengiriman data. *Node* berwarna biru (9) sebagai *node* tujuan yang berperan untuk menerima data. Dan *node* berwarna merah (3) dan (7) sebagai *node black hole* yang berperan sebagai serangan.

4.1.32 Hasil Pengujian Parameter 100 *Node* Terdapat *Black Hole*

Berikut adalah hasil parameter pengujian pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ awk -f parameter.awk hasil/black
hole/hasil-bh.tr
========================
   Parameter Pengujian
========================
GeneratedPackets = 239
ReceivedPackets = 0
Packet Loss =  239
Packet Delivery Ratio = 0%
Average End-to-End Delay = 214.755 ms
Total Dropped Packets = 239
```

Gambar 4.32 Hasil parameter pengujian 100 *node* terdapat serangan *black hole*

Gambar 4.32 merupakan tampilan pada terminal ubuntu hasil simulasi jaringan WSN dalam kondisi normal menggunakan aplikasi NS2. Pada gambar di atas menujukan hasil *packet delivery ratio* 0%, *packet loss 239* dan *average end to end delay* 214.755 *ms.*

4.1.33 Hasil Pengujian Deteksi Serangan *Black Hole* 100 *Node*

Berikut adalah kegiatan *node* pada kondisi terdapat *black hole.*

```
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$ perl ujiSVM.pl hasil/blackhole/h
asil-bh.tr
JML PAKET : 25834
PAKET BLACKHOLE : 19120
PAKET NORMAL : 6714

Prosentasi Paket Blackhole : 74.0109932646899%
DONE
(base) skripsi@awit:~/Desktop/MINE/Awiiit/mine$
```

Gambar 4.33 Hasil pengujian SVM *black hole* 100 *node*

Pada Gambar 4.33 di atas menjelaskan jumlah paket sebanyak 25834, yang dibuang oleh *black hole* sebanyak 19120, sisa paket normal sebanyak 6714, persentasi paket *black hole* sebanyak 74.0109%.

4.2 Data Hasil Pengujian Deteksi

Berikut adalah data hasil pengujian deteksi serangan *black hole*:

Tabel 4.1 Hasil pengujian dengan parameter

No	Aktivitas	Jumlah Node	Paket Loss	PDR (%)	Average Delay (ms)
1	Normal	10 *node*	1	99.5816	202.429
2	Serangan	10 *node*	239	0	172.524
3	Serangan	20 *node*	239	0	172.817
4	Serangan	30 *node*	239	0	172.526
5	Serangan	40 *node*	239	0	172.53
6	Serangan	50 *node*	239	0	214.699
7	Serangan	60 *node*	239	0	172.979
8	Serangan	70 *node*	239	0	214.729
9	Serangan	80 *node*	239	0	214.734
10	Serangan	90 *node*	239	0	214.783
11	Serangan	100 *node*	239	0	214.755

Tabel 4.1 menampilkan data hasil pengujian parameter pada simulasi jaringan WSN dengan *node* normal dan *node* terdapat serangan *black hole*. Data didapatkan dari 11 simulasi dengan jumlah *node* yang berbeda. Skenario pengujian yang dilakukan dengan menambahkan jumlah *node* pada tiap simulasi. Terdapat dua *node* serangan, posisi *black hole* berada pada *node* (3) dan *node* (7). Pada table 4.1 menunjukan hasil pengujian dengan 3 parameter *Quality of Service* (QoS) yaitu nilai *packet loss*, *packet delivery ratio* (PDR) dengan satuan (%), dan *delay* dengan satuan (ms).

Tabel 4.2 Hasil pengujian deteksi serangan *black hole* variasi *node*

No	Aktivitas	Jumlah *Node*	Jumlah *Node Black hole*	Persentasi Paket *Black hole*
1	Normal	20 *node*	0	0%
2	Serangan	10 *node*	2 *node*	94.9%
3	Serangan	20 *node*	2 *node*	91.1%
4	Serangan	30 *node*	2 *node*	89.4%
5	Serangan	40 *node*	2 *node*	86.8%
6	Serangan	50 *node*	2 *node*	84.3%
7	Serangan	60 *node*	2 *node*	82.8%
8	Serangan	70 *node*	2 *node*	79.2%
9	Serangan	80 *node*	2 *node*	78.1%
10	Serangan	90 *node*	2 *node*	73.5%
11	Serangan	100 *node*	2 *node*	74%

Tabel 4.2 menampilkan data hasil pengujian simulasi jaringan WSN dengan *node* normal dan *node* terdapat serangan *black hole*. Data didapatkan dari 11 simulasi dengan jumlah *node* yang berbeda. Skenario pengujian yang dilakukan

dengan menambahkan jumlah *node* pada tiap simulasi. Terdapat dua *node* serangan, posisi *black hole* berada pada *node* (3) dan *node* (7). Pada table 4.1 menunjukan persentasi paket normal dan paket *black hole* pada setiap simulasi, jika diperhatikan, persentase paket *black hole* menurun dengan bertambahnya jumlah *node*.

BAB 5 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap hasil pengujian yang telah dilakukan sebagai berikut:

5.1 Kesimpulan

1. Pada hasil pengujian, simulasi dengan kondisi normal pada *packet loss* hanya bernilai kecil (1), sedangkan simulasi terdapat *black hole* sebanyak (239) 100%. Simulasi dengan kondisi normal pada *packet delivery ratio* sebanyak 99.5816%, sedangkan simulasi terdapat *black hole* bernilai 0%. Simulasi dengan kondisi normal pada *average end-to-end delay* sebesar 202.429 ms, sedangkan simulasi terdapat *black hole* sekitar 172.524 hingga 214.979.
2. Algoritma SVM dapat diterapkan pada klasifikasi serangan *black hole* dengan menerapkan perhitungan kernel polinominal dengan hasil yang akurat.
3. Jumlah *node* mempengaruhi persentase paket *black hole,* jika *node* semakin tinggi maka persentase paket *black hole* menurun.

5.2 Saran

Selanjutnya dapat dilakukan penelitian lanjutan seperti melakukan pencegahan terhadap serangan *black hole* pada jaringan WSN.

DAFTAR REFERENSI

Asma, A., A, Hanan., and Izzeldin O., 2015. *AODV ROUTING PROTOCOL WORKING PROCESS.* Volume 10, Number 2, March 2015

Dorri, A. and Kamel, S.R., 2015. Security Challenges in Mobile Ad Hoc Networks: A Survey. *International Journal of Computer Science & Engineering Survey*, [online] 6(1), pp.15–29. Available at: <http://www.airccse.org/journal/ijcses/papers/6115ijcses02.pdf>.

Gurjot, S. and Jagdeep S., 2013. *Prevention of Black Hole Attack in Wireless Sensor Network using IPSec Protocol.* Volume 4, No. 11, Nov -Dec 2013

Huanan, Z., Suping X., Jiannan, W., 2021. *Security and application of wireless sensor network.* Procedia Computer Science, Volume 183, Pages 486-492, ISSN 1877-0509

Kalkha, H., Satori, H., Satori, K., 2019. *Preventing Black Hole Attack in Wireless Sensor Network Using HMM.* Procedia Computer Science, Volume 148, Pages 552-561, ISSN 1877-0509

Mehndi, S. and Naveen, Kumar, G., 2016. *Black Hole Attack Detection in Wireless Sensor Networks Using Support Vector Machine. 3(5): 48-52.*

Miriam, Carlos, M., Ernesto, López, M., dan Mario S., 2016. *Wireless Sensor Networks Formation: Approaches and Techniques.* Volume 2016, Article ID 2081902, 18 pages

Nugroho, A.S., Witarto, A.B., Handoko, D., *Application of Support Vector Machine in Bioinformatics.* Proceeding of Indonesian Scientific Meeting in Central Japan, December 20, 2003, Gifu-Japan

Rohankar, R., Bhatia, R., Shrivastava, V., Sharma, D.K. 2012. *Performance Analysis of Various Routing Protocols (Proactive and Reactive) for Random Mobility Models of Adhoc Networks*. 1st Int'l Conf. on Recent Advances in Information Technology

Tseng, FH., Chou, LD. & Chao, HC. 2011. *A survey of black hole attacks in wireless mobile ad hoc networks*. Hum. Cent. Comput. Inf. Sci. 1, 4 (2011).

Umashankar, G. and Jayaram, P., 2017. *A Study on Black Hole Attack in Wireless Sensor Networks.* ISSN : 2321-4546, Vol 5